Method for the compilation of bus packets for isochronous data transmission via a data bus, and apparatus for carrying out the method

The invention relates to a method for the compilation of bus packets for isochronous data transmission via a data bus. The invention furthermore relates to an apparatus for carrying out the method. The apparatus may be, in particular, part of a bus interface for the connected data bus.

Prior art

The invention is based on a method for the compilation of bus packets for isochronous data transmission via a data bus of the generic type of the independent Claim 1. For quite a long time now the convergence of the product sectors of consumer electronics (hifi, video, audio) and personal computing has been trumpeted under the catchword multimedia and has actually been propelled by many manufacturers from both camps. The merging of the two product sectors means that work concerned with the subject of data exchange between the equipment of the different product sectors or else between the equipment within one product sector is becoming more and more significant. This is also apparent from the efforts for standardization with regard to this subject, which are already well advanced. Specifically, the so-called IEEE 1394 serial bus already provides an internationally standardized and very widely accepted bus for data exchange between terminals from both product groups. The precise designation of the aforementioned standard is: IEEE Standard for high performance serial bus, (IEEE) STD 1394-1995, IEEE New York, August 1996.

The invention that is to be described here is concerned with the so-called isochronous data transfer within the abovementioned bus system. In this connection isochronous means that data to be transmitted arise regularly at a data source, the data also arising with

approximately the same size each time. Examples of such data sources are video recorders or camcorders, audio devices such as CD players or DAT recorders, and also DVD players or videophone devices, etc. An international standard has been specially developed for this application of isochronous data transmission. The precise designation of this standard is: IEC International Standard 61883 "Consumer Audio/Video Equipment Digital Interface, 1st edition 1998". The first part of this standard describes the general data packet format, the data bus management and the connection management for audio visual data. General transmission rules for control commands are likewise defined.

Very frequent application relates to the transmission of MPEG2-coded video or audio data. The data are transported via the bus in packets, as already mentioned. In this case, the following structure is provided in the abovementioned Standard IEC 61883: the data generated in the data source are divided into so-called data source packets having a defined size. For MPEG2 video data transmission, for example, the standard stipulates that a data source packet is composed for example of 8 data blocks of identical size. In this case, the data block size can be programmed. It may be between one and 256 quadlets, where a quadlet corresponds to a combination of 4 data bytes. The data source packets are transmitted in one or more bus packets in accordance with the IEC 61883 Standard. A bus packet has a so-called isochronous data format header in addition to the elements of bus packet header, useful data field and CRC checksum field. The said isochronous data format header is designated as CIP header (Common isochronous packet) in the abovementioned IEC 61883 Standard. It defines the data format for isochronous data transmission, which is described in detail in the Standard and will be explained in more detail below. This isochronous data format header is called CIP header below. This CIP header is added to the beginning of each bus packet after the bus packet

header. This then ensures that the station which receives the transmitted bus packet can evaluate the data in the correct manner.

Although the CIP header largely remains constant throughout the isochronous data transmission, it must nonetheless be newly updated in one section (DBC entry). Added to this, however, is the fact that during isochronous data transmission, the said CIP header has to be present twice in the bus interface, since, on the one hand, a completely compiled bus packet can be sent onto the bus, while at the same time new data are provided by the application process, a new CIP header having to be created for these new data.

Owing to this difficulty, we initially considered a solution internally for the compilation of bus packets in which two separate special registers are provided for the CIP headers. The useful data of the respective bus packets are provided in a buffer memory. If a packet is to be sent via the bus, then a selection unit must read the correct CIP header from one of the two special registers and transfer it to the data transmitting unit at the correct point in time and then also fetch the associated data from the buffer memory and attach them to the CIP header.

The object of the invention is to simplify the solution described above, to be precise in such a way that the selection logic unit for joining together the CIP header and the associated useful data can as far as possible be omitted.

The invention achieves this object in such a way that, when the isochronous data transmission is set up, it writes the generated CIP header only to one special register and, in addition, also to the buffer memory for the useful data, in which case the useful data of the bus packet are subsequently attached to this CIP header in the buffer memory (see Claim 1). What is achieved as a result of this is that, for the transmission of the data via the bus, the data transmitting section only has to

access the buffer memory for the useful data, where CIP header and useful data are stored contiguously in the correct order. The data transmitting section thus obtains the data to be transmitted only via the buffer memory. A selection logic unit which determines the special register from which the CIP header has to be taken and the memory area of the buffer area from which the useful data have to be attached can be omitted.

Further improvements of the method are possible by virtue of the measures evinced in the dependent claims. According to Claim 2, the CIP header may contain a comparison value for data counting. This value must be updated for each bus packet. This is done in such a way that when the data of a bus packet are written to the buffer memory, the data are counted and, at the end, the comparison value, determined in this way, for the data count is updated in the CIP header, which is entered in the special register, and the updated CIP header is copied to the buffer memory at the next free location for a bus packet. The data of the next bus packet would then be attached in turn to this CIP header. Consequently, the useful data for the next bus packet are again stored contiguously in the buffer memory and they can be transported from there contiguously to the data transmitting section of the bus interface.

In this connection, it is advantageous if the data are counted in units of data blocks and the comparison value for the data count in the CIP header relates to the first data block in the respective bus packet. As a result, the solution then conforms to the abovementioned IEC 61883 Standard, which also stipulates that the comparison value DBC in the CIP header relates in each case to the first data block of a bus packet.

The following measures which specify the way in which the corresponding object of the invention is achieved are advantageous for an apparatus for carrying out the method according to the invention (see Claim 6). The apparatus comprises a buffer memory for the data of

bus packets. Furthermore, the apparatus comprises a memory management unit and a special register for a CIP header of a bus packet. The apparatus furthermore has initialization means which, when the isochronous data transmission is set up, copy the corresponding CIP header for the first bus packet both to the special register and to the buffer memory.

The CIP header for the isochronous data transmission to be set up is preferably prescribed by the application process in the transmitting station.

Also advantageous are the measures according to Claim 8, where it is defined that the apparatus furthermore has a data block counter, by which the data blocks of the isochronous data transmission are counted and whose counter reading at the corresponding point in time specifies the comparison value for the data count, which is entered into the special register in which the CIP header for the isochronous data transmission was stored during initialization. Furthermore, provision is made for the respective updated CIP header to be copied to the buffer memory, with the result that the correct CIP header is directly available again in the buffer memory for the next bus packet to be transmitted.

Drawings

Exemplary embodiments of the invention are illustrated in the drawings and are explained in more detail in the description below. In the figures:

Figure 1 shows the structure of a plurality of successive bus packets for isochronous data transmission, and

Figure 2 shows a block diagram of the apparatus according to the invention.

Exemplary embodiments of the invention

Figure 1 shows an exemplary sequence of transmitted bus packets. In the example shown, it is assumed that MPEG2-coded video data are to be transmitted

during the isochronous data transmission. For this case, the IEC 61883 Standard provides for 8 data blocks with MPEG2 video data to be transmitted per data source packet. The size of the data blocks is specified in units of quadlets in the abovementioned standard. The data block size can be programmed; to be precise, all values between one and 256 quadlets are possible. For the transmission of MPEG2 video data, the IEC 61883 Standard provides for a data block to have a size of 6 quadlets. Furthermore, it is assumed that in each case 8 data blocks are transmitted in a 1394 bus packet. This is possible according to the abovementioned standard and, in this case, all the data blocks of a data source packet can be completely transmitted in one bus packet. Figure 1 shows an exemplary sequence of transmitted bus packets. The first transmitted bus packet is illustrated at the top in Figure 1 and the second transmitted bus packet is correspondingly illustrated at the bottom in Figure 1. The precise structure of a bus packet for isochronous data transmission is specified in the abovementioned IEC 61886 Standard. Therefore, for the disclosure of the invention, reference is also expressly made to this standard.

In Figure 1, the reference numeral 10 designates the header of the bus packet. It contains the details regarding the data field of the isochronous data packet, to be precise in a number of bytes, and also further information, but this need not be discussed in any more detail below. The header 10 of the bus packet is followed by a data field. The latter extends through the area 11-19. At the end of the bus packet there also follows an area 20, in which a CRC check word is stored. A so-called CIP header is always provided at the beginning of the data field of a bus packet. CIP is the abbreviation of "Common isochronous packet". The CIP header contains a series of information items which describe the isochronous data transfer. Thus, e.g. an identification number SID of the data source is contained therein.

Furthermore, it stipulates the size of the subsequent data blocks in the bus packet. Likewise, a detail FN (fraction number) is also contained, which specifies the number of data blocks into which a data source packet is divided. As already mentioned, there are always 8 data blocks per data source packet in the case of MPEG2 video data. A further detail QPC (quadlet padding count) relates to how many padding quadlets are attached at the end of the data source packet in order to guarantee that the latter is divided into data blocks of the same size. Furthermore, an information item SPH (source packet header) is provided, which specifies whether a header for the data source packet is likewise also provided in the bus packet. Furthermore a DBC value (data block counter) is also provided. This value specifies which data block is the first data block in the bus packet referring to all the transmitted data blocks during the isochronous data transmission. Therefore, all the data blocks are consecutively numbered individually. This value practically constitutes a comparison value which can easily be used to check whether a bus packet has not been received. To that end, the received data blocks are all counted up in the receiver station. Each time a new bus packet is received, the DBC value contained therein is compared with the counted comparison value. Only if both values correspond have all the data blocks been received and no bus packet has been lost. Further information items in the CIP header include an FMT entry (format ID). This entry can be used to signal that the bus packet contains no data at all and is a so-called dummy packet. An FDF entry (format depending field) may also be defined, this being mentioned only for the sake of completeness, and also an SYT entry, which comprises a time specification for the bus packet.

The data blocks DB0-DB7 for the first data source packet SP0 then follow in the subsequent areas 12-19. The entry 0 in the data area 11 is intended to indicate that the DBC value for this first bus packet is set to the

value 0, which is synonymous with the fact that the first data block in this bus packet has the number 0. The DBC value is automatically set to this value during the initialization of the isochronous data transfer. This will be explained in more detail below. This must, of course, also be taken into consideration for the comparison count. Therefore, the comparison count likewise begins at 0.

The next bus packet again contains 8 data blocks. In this case, they are the 8 data blocks DB0-DB7 of the second data source packet SP1. This may also be followed by further bus packets which are likewise constructed in the manner illustrated.

The relevant parts of a bus interface for the invention are illustrated in Figure 2. These components are parts of a data link layer circuit within the IEEE 1394 bus interface. The reference numeral 30 designates an I2C interface, to which an I2C bus 38 is connected. Via the I2C interface, the IEEE 1394 bus interface can be configured e.g. for isochronous data transmission. The necessary control data are prescribed by an application process via the I2C bus 38. The I2C interface 30 is connected via an internal bus 41 to further components of the bus interface. The reference numeral 32 designates a buffer memory for the data exchange. This buffer memory 32 is managed by the memory management unit 31. In other words, the memory management unit 31 divides the memory in such a way that the incoming and outgoing data are correctly forwarded to the components which each access the memory. The entire address control thus takes place with the aid of this memory management unit 31. It also serves as a bus master for the internal bus 41 and allocates it to the connected units by time division multiplexing.

Furthermore, an AV transceiver unit 33 is connected to the internal bus 41. This unit is in turn connected to a data bus 39, via which all the incoming and outgoing data are relayed to and from the

application. The AV transceiver unit 33 also comprises a DB counter 37. This DB counter counts up all of the data blocks received from the application. In accordance with the IEC 61883 Standard, this counter is an 8 bit counter.

As a further component, a register unit 34 is also connected to the internal bus 41. The said register unit also contains the already mentioned special register for the CIP header.

Further components which are also connected to the internal bus 41 relate to a data transmitting circuit 35 and a data receiving circuit 36. These circuits are connected to the physical layer IC of the 1394 bus interface. Their function, in the case of the transmission of data via the 1394 bus, is to take the corresponding bus packet data from the buffer memory 32 and forward them in the correct order to the physical layer IC. A further task of the data transmitting unit 35 is to perform the CRC check and to attach the corresponding CRC check data at the end of a bus packet. In the case of the 1394 bus, a separate CRC check is provided for the data in the 1394 header of the bus packet. This is likewise handled by the data transmitting unit 35. The data receiving unit 36 has corresponding tasks, namely CRC checking of a received bus packet separately for the 1394 header and for the useful data, and the extraction of the useful data from the bus packet and the forwarding of these data to the buffer memory 32.

The method of operation of the apparatus will now be explained in more detail below. If an isochronous data transfer is requested by the application process, the following takes place. The bus interface is initialized via the I2C interface 30, all the units being prepared for the isochronous data transmission. In particular, the CIP header for the isochronous data transmission, which header is prescribed by the application with the corresponding values, is entered on the one hand into the special register 38 and on the other hand at the first free location in the buffer memory 32 for a bus packet.

It should be mentioned here that the DPC comparison value in the CIP header is set to 0 on account of the initialization. Equally, the counter reading of the DB counter 37 is also reset to 0 as a result of the initialization. Furthermore, the entry for the 1394 header is written to the 1394 header special register 39. This entry depends on the entries in the special register for the CIP header 38. Since the 1394 header does not change throughout the isochronous data transmission, it is not absolutely necessary to transfer this 1394 header simultaneously to the buffer memory 32 as well. Specifically, it is possible to adopt the corresponding 1394 header from the special register 39 each time a bus packet is transmitted. After the 1394 bus interface has been set up for the isochronous data transfer requested, the useful data are supplied by the application via the bus 41. The AV transceiver unit 33 forwards the incoming data in corresponding memory words to the buffer memory 32. The integrated DB counter 37 counts up the data and is incremented each time a complete data block has been forwarded to the memory. The size of the data block is, after all, entered in the special register 38 and the DB counter 37 was set accordingly during the initialization process. After 8 data blocks have then been written to the buffer memory 32, the DB counter 37 outputs a control signal, whereby its current counter reading is transferred to the special register 38, to be precise at the location for the comparison value DBC. At the same time, this signal informs the memory management unit 31 that it should copy the updated CIP header in the special register 38 to the next free location for a bus packet in the buffer memory 32. Afterwards, further useful data can then be written to the buffer memory 32 via the AV transceiver 33. At the same time as new data are being written in, however, the data of the preceding bus packet can be output onto the 1394 bus via the data transmitting unit 35 and the physical layer IC. The memory management unit 31 allocates the internal bus 41 to the various

components by time division multiplexing. In this case, the internal bus 41 is designed in such a way that it can satisfy the bandwidth requirements of the individual components. After all, there is the added fact that via the data receiving unit 36, too, they may enter requirements for forwarding received data into the buffer memory 32, so that the bandwidth requirements of the latter must also be satisfied.

The fact that the CIP header for a bus packet to be transmitted resides in each case at the beginning of the assigned memory area for this bus packet in the buffer memory 32 ensures that when the bus packets are transmitted, first of all access can be made to the special register 39, where the 1394 header of the bus packet is stored, and then all of the further data can be taken from the buffer memory 32. This operation is simple to carry out and a relatively complicated switching logic arrangement is not necessary for this purpose.

Various adaptations and modifications of the exemplary embodiments described are possible. The structure with the various internal bus lines and bus lines provided for the external components, as described, may be chosen differently. Parts of the explained apparatus may also be realized by software. The invention is not restricted to use with the IEEE 1394 bus mentioned. It can also be used for other wire-based bus systems or else for a wire-free bus system.